International Journal of Computer Science
and Engineering (IJCSE)
ISSN(P): 2278-9960; ISSN(E): 2278-9979
Vol. 3, Issue 1, Jan 2014, 7-18
© IASET

International Academy of Science,
Engineering and Technology
Connecting Researchers; Nurturing Innovations

# BATGA: A GENETIC ALGORITHM WITH DAMPED-ACCELERATION BASED ADAPTIVE TERMINATION MECHANISM

**TRIDIB R. SARMA**

Associate Professor, School of Management Sciences, Tezpur University, Tezpur, Assam, India

## ABSTRACT

This paper presents a design for a Damped-Accelerator based termination for Genetic Algorithms. The necessity is felt to expand the sampling space within the search space when the search space size varies.

The concept is tested on the Resource-Constrained Project Scheduling Problem (RCPSP), which is an acknowledged NP-hard problem. By testing a gradually evolving set of termination mechanism, this paper finally conceptualizes a termination criterion incorporating a combination of exponentially accelerating but logarithmically damped mechanism. Mathematical proof of the concept is kept outside the scope of this paper. Experiments on benchmark test instances have shown results comparable with the best in literature.

**KEYWORDS:** Genetic Algorithms, Damped, Acceleration, Resource Constrained Project Scheduling

## INTRODUCTION

For any combinatorial problem the total number of feasible solutions – within the search space – is an indicator of its complexity. In case of use of exact method for finding the optima, the amount of time required would be directly proportional to the size of the search space. On the other hand, heuristics can be designed to provide an 'acceptable' solution – not necessarily the exact 'best'. But this would have its own inherent drawbacks. A very effective and strong heuristic method for optimizing combinatorial problem is Genetic Algorithms. Since its conception, Genetic Algorithms has been widely utilized for optimizing varieties of such problems. We take up one such problem, the RCPSP, for optimizing it by the usage of Genetic Algorithms. In doing so, a novel concept shall be placed for furthering the continuous evolution of Genetic Algorithms itself. Sarma (2004) had presented a roadmap for combination of meta heuristics, which would teach the algorithm to be adaptive in optimizing the RCPSP.

The main focus of this paper would be on the development of a termination criterion for applying Genetic Algorithms in situation(s) having a variable search space size. For testing the concept, RCPSP is taken as the case. Without going into complex statistical checks, the effectiveness, accuracy and efficiency of the proposed termination criterion is tested using a few simple metrics developed specifically for the purpose.

## THE RCPS PROBLEM

Resource Constrained Project Scheduling Problem (RCPSP) is a specialized version of the Resource Constrained Scheduling Problem (RCSP). Study of RCS initially started with job sequencing in Shop-Floor, and allocating finite number of machines, operators, etc. Scheduling problems tend to be difficult, not just in theory, but in practice as well. Applegate and Cook (1991) remarked that the job shop problem is not only NP-hard, it also has the well-earned reputation of being one of the most computationally stubborn combinatorial problems to date. In their book, Muth and Thompson (1963) introduced a ten machine, ten job problem that took the Operations Research community more than two decades to arrive at a plausible solution set.

The Resource-Constrained Project Scheduling Problem (RCPSP) consists of a set of tasks, and a set of finite capacity resources. Each task puts some demand on the resources. A partial ordering of these tasks is given specifying that some tasks must precede others. Within the limited quantity of available resource(s), one is faced with the problem of optimizing delivery time of the project. The aim is to optimize allocation of constrained resource(s) amongst the competing tasks in hand, at the shortest possible time. In other words, one needs to shorten *makespan* of the project – all the time staying within limits imposed by availability constraint(s) of resource(s) as well as precedence constraint(s) of tasks.

The RCPSP may be described as follows:

- Given
  - A set of activities that must be executed,
  - A set of resources and their capacity limitations to be utilized,
  - A set of quality objectives by which one may judge performances
- All within the non-negotiable boundary of a (set of) constraints
- What would be the best way of assigning the resources to the activities within the constrain limitations so as to achieve the best objective(s) of completion of the Project.

As adapted from Crawford (1996) the RCPSP can be depicted:

Given: a set of tasks, T,

a set of resources, R,

a capacity function, $C: R \rightarrow N$,

a duration function, $D: T \rightarrow N$,

a utilization function, $U: T \times R \rightarrow N$,

a partial order, P on T, and

a deadline, d.

**Objective**

To achieve Min $\sum D$, by assignment of start times $S: T \rightarrow N$

**Subject to the Constraints**

- **Precedence Constraints:** If $t_1$ precedes $t_2$ in the partial order P, then $S(t_1) + D(t_1) \leq S(t_2)$
- **Resource Constraints:** For any time x, let running(x) = $\{t|S(t) \leq x < S(t) + D(t)\}$.

  Then for all times x, and all $r \in R$, $\sum_{t \in running(x)} U(t,r) \leq C(r)$.
- **Temporal Constraints**: For all tasks t : $S(t) \geq 0$, and

$$S(t) + D(t) < d$$ **Relationship 1**

For arriving at the optimal solution, the problem may be tackled from a number of angles, such as

- Formation of task sequence,

- Allocation of scarce resource to the tasks,
- Delimitation of time-windows of the tasks,
- Defining and designing of alternative mode of execution, etc

For our optimal solution, the present work is an attempt to formulate the task sequence within prescribed time duration of individual tasks, and allocate scarce resources to the needy tasks, and finally provide not just one 'optimal' solution but a 'optimal set of solutions'. The 'set' is provided from a reality point of view. In case of emergency, the Project Manager would have feasible alternatives if a specific solution becomes unusable.

## GENETIC ALGORITHMS APPROACHES FOR OPTIMIZING THE RCPSP

Since the conception of RCPSP, a number of optimization methods were propounded – both exact as well as heuristic. Methods ranged from Linear Programming to Network based methods, from exact mathematical modeling to probability based heuristic modeling methods. Yang et.al.(2001) had made an extensive study of different approaches. Amongst contemporary heuristics, the most extensively used is Genetic Algorithms - individually and in combinations with other heuristics, on all the variations of RCPSP.

More than two decades ago, the Committee on the Next Decade of Operations Research, CONDOR Report (1988), singled out Tabu Search, Simulated Annealing, and Genetic Algorithms as 'extremely promising' optimization methods for the years to come. The foresightedness of this report is vindicated by the fact that these three approaches are still being widely used – albeit with extensive adaptations, evolutions and modifications. Application of these methods for RCPSP optimization has thrown up twofold results – better optimized solutions as well as faster algorithms.

Kolisch and Hartmann (2006), in the updated report have given short comparative description of application of heuristics and metaheuristics by different researchers for the RCPSP. On their list, most of the work that produced 'good' results are metaheuristics, where TS, SA and GA tops the list. Robustness of other metaheuristics, esp. swarm-optimization, ACO, etc have also been established for optimization of the RCPSP in recent years.

## PROPOSAL AND IMPLEMENTATION OF THE ADAPTIVE TERMINATION ALGORITHM

In general, the Genetic Algorithms approaches for optimizing RCPSP (or for that matter, any application area) primarily deals with the major components of the algorithm, viz. Crossover and Mutation. The other components, viz. Selection, Termination, etc even though dealt with, but are generally given lesser degree of importance, as evident from literature.

To avoid a perpetual Genetic Algorithm, it has to be terminated once certain criteria are met. We experimented with termination based on two classical methodologies, viz. Fixed Generations Termination and Fitness-Deviation Termination. Thereafter, the concept of Adaptive Termination algorithm is placed for implementation, and compared with results obtained from the others mentioned above.

### Concept of the Adaptive Termination

Each Project has its own set of tasks and resource types. Moreover, Projects differ in their number of possible solutions, or the search space. This we term as Complexity level. The Complexity level of a Project increases by direct (linear or exponential function based) proportion to the precedence constraint of Project tasks. Keeping this in mind we designed an algorithm that would be adaptive to these three Project parameters for deciding the termination criteria.

The adaptive algorithm is designed by evolving it to a final stage through gradual incorporation of parameters in three stages.

**The Basic Structure**

The proposed Adaptive Termination algorithm would be a function of three project parameters, viz.

Project length or the number of tasks,

Number of resources under constraint, and

Complexity level.

Combined, they arrive at the number of generations the GA would run, which we shall term as *MaxGen*, as given in Relationship 2

$G = f[(T, R,) C]$ **Relationship 2**

where, G is number of generations, MaxGen

T is number of tasks,

R is number of constrained resources, and

C is Complexity level

We group up the first two factors and term the group as Project Factor (P), and rework the last factor as Complexity Factor (C). Thus as a basic structure, we get G as a function of P and C, or

$G = f[P, C])$ **Relationship 3**

where $P = f[T, R]$

**Adaptive Termination 1 (Adapterm1)**

Initially the algorithm is based on two assumptions –

- The Project is ‘not complex’ hence doesn’t require emphasis on its complexity level, and
- Is dependent only on the Project Factors.

With such assumptions, we calculate G by marginalizing C, i.e. keep it at 1, to get Relationship 4

$G = f[P]$, with $C = 1$ **Relationship 4**

where $P = f[T, R]$,

For example, if the project has 60 tasks and utilizes 4 types of constrained resources then the Genetic Algorithm is processed for 240 generations.

**Adaptive Termination 2 (Adapterm2)**

Next we remove the above assumption regarding ‘Complexity’ of the Project, and introduce C through combination of functions.

$G = f[T, R, f'(\theta)]$ **Relationship 5**

where $\theta$ is the indicator of search space, i.e. the complexity level, and

C = $f'(\theta)$

The segment $f'(\theta)$ is combined with Relationship 4 to dampen the rate of change (or velocity) of complexity. This is the second stage of evolution of the proposed Adaptive Termination.

Logarithmic functions being a good damper, and we propose to utilize the same.

**Adaptive Termination 3 (Adapterm3)**

By use of Logarithmic functions, Relationship 5 would damp the complexity level to a very low value, almost making the rate of growth flat. The exponential nature of complexity is brought back and the Termination criteria allowed to accelerate, but under damping by a logarithmic function. Thus the algorithm is allowed to evolve further.

The conceptual mechanism of this Damped-Acceleration due to combination of Logarithmic and Exponential function on an exponential set of data is illustrated in Figure 1. .

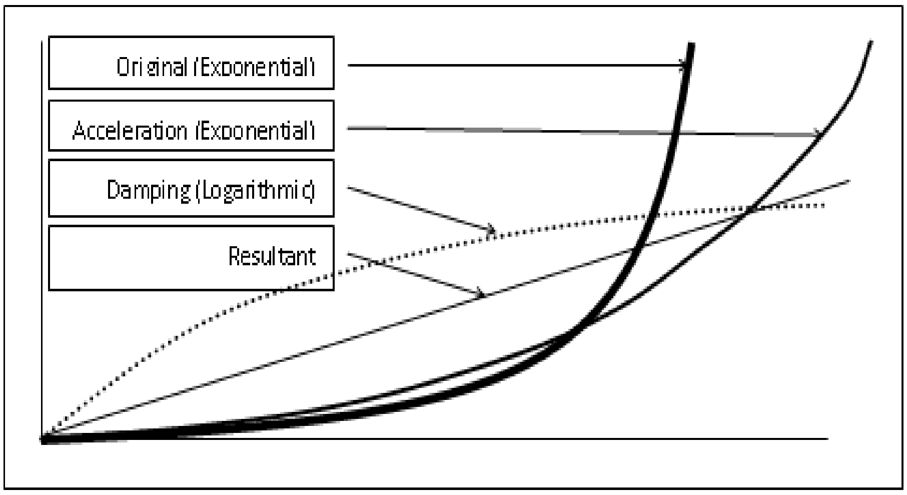

**Figure 1: Concept of the Damped-Acceleration Mechanism**

We introduce a combination of exponential and logarithmic functions for addressing the characteristics of project complexity. Exponential function is proposed to respect velocity of search space increase, and Logarithmic function would try to control it, as depicted in Relationship 6

$C = f'(\theta) = f''[\beta(\log (b, \theta, \phi))]$ **Relationship 6**

where

$\beta$ is the base of an exponential function for C, exponentially raised to a function of $\theta$ and $\phi$,

bis base of the logarithmic operation and

$\phi$is a limiting number which depends on the search space

For implementation, the relationships for Adaptive Termination are made specific as depicted below

| | | |
|---|---|---|
| i) | Adaptive Termination 1 (*AdapTerm1*) | |
| | G = T * R | Relationship 7 |
| ii) | Adaptive Termination 2 (*AdapTerm2*) | |
| | G = [T * R]*INT[$\log_b$M] | Relationship 8 |
| iii) | Adaptive Termination 3 (*AdapTerm3*) | |
| | [INT($\log_b$M) – ($\phi$-$\omega$)] | |
| | G = [T * R] * $\beta$,for M >= $10^{\phi}$ | Relationship 9(a) |
| | = [T * R] * INT[$\log_b$M],for M < $10^{\phi}$ | Relationship 9(b) |
| | Where<br>G: *MaxGen*<br>T: Project length<br>R : Constrained Resources<br>M: *MaxSdl,*<br>$\beta$ : Base of the exponential function,<br>b : Base of the logarithm used, and<br>$\phi$ : A limiting factor<br>$\omega$ : A small non-negative integer. | |

The mathematical proof of the proposed algorithm is kept outside scope of this work. Experimentation was carried out by computational application of the algorithm using various combinations of the parameters using a carefully structured DoE (Design of Experiments).

## METHODOLOGY FOR RESULTS ANALYSIS

The Algorithm is validated from three angles – *effectiveness*, *accuracy* and *efficiency*. Simple statistical tests are devised for testing combinations of the aforementioned parameters.

- To validate effectiveness of the algorithm and compare with results of other researchers, we use Percentage Average Deviations (PAD) as devised by Kolisch and Hartmann (2006) in their methodology.

PAD= {∑ Makespan [Test] – ∑Makespan [Reference]}/ {∑ Makespan [Reference]} * 100 % **Relationship 10**

For our experimentation, we have taken published PSPLIB information as Reference. The results are considered better as PAD keeps reducing, and approaches zero.

- Average Performance Level Variation (APLV), which we have specifically defined, measures a change in performance between the two combinations. This test provides accuracy of the algorithm in achieving optimization.

APLV(this, other) = [$\text{PAD}_{other}$ –$\text{PAD}_{this}$ ] / [$\text{PAD}_{other}$]* 100% **Relationship 11**

A positive value implies an improvement in performance due to ‘*this’* combination as compared to some ‘*other’* combination.

- To validate efficiency of a selected combination for the algorithm, we define and calculate an *Efficiency Index* (EI) for each test instance. The *Efficiency Index* is modeled in line of designing the Decibel scale that expresses the magnitude of a physical quantity (usually power or intensity) relative to a specified or implied reference level. A similar logarithmic index is the Richter scale, used for quantifying the magnitude of earthquakes.

This *Efficiency Index* is an indication of effectiveness of the method in reaching a level of accuracy within the sample area of the total search space. *Efficiency Index* for the $i^{th}$ instance within a specific data-set,

$\text{EI}_i$ = $\text{Log}_{10}$ {[Makespan[Reference]$_i$ / Makespan[Test]$_I$ ]/ [Sample proportion $_i$]} **Relationship 12** where,

Sample Proportion $_i$ = *MaxPop* * *MaxGen* $_i$ [as per Termination Criteria used] / *MaxSdl* $_i$ **Relationship 12a**

Efficiency (of a combination) is considered better if *Efficiency Index* is higher as compared to that of another combination for the same instance. This is an index at the instance level, and is not to be averaged out over the total data-set.

For implementation, well-established adaptations of Genetic Algorithms components for RCPSP were made, on which our proposed termination criteria was tested. For generation of solutions, we used a serial SGS, which assures that all solutions generated are valid and within the search space. A combination algorithm based unique number, which rides on the sequence of the tasks, and the project's makespan is utilized for Selection.

This unique number is used as Fitness Function for multiple purposes throughout the algorithm. The robust mechanism of Precedence-Set Crossover (PSX) is adopted from Alcaratz - Maroto for binary combination. Because of possibility that mutation would generate infeasible solution, we avoid this component. But utilize immigration as a unary combination component to introduce diversity.

Internationally accepted standard benchmark instances provided by Kolisch and Sprecher (1996) for evaluation of the RCPSP are being used to test the proposed termination criteria of the Genetic Algorithm. These are available from the digital library called PSPLIB, and is widely acknowledged in literature for the purpose. From PSPLIB, we employed the standard SMFF *(Single Mode, Full Factorial)* sets – J30 and J60. These consist of (48 X 10 =) 480 project instances each, of Project lengths indicated by the number in the labels.

## EXPERIMENTAL RESULTS

An Experiment Serial Number, ESN, labels each of the experiment results depicted in the tables below for reference required subsequently. The first set of experiments is in 'primitive' combination, where the parameters are set as collected from literature. This is done to allow the results to be 'crude'. Crossover is done using Mid-Point PSX mechanism. This shall be changed to Random-Point PSX mechanism as we go into testing of the Adaptive Termination criteria. By such gradual changing and tuning of parameters, these results are brought to within the best amongst Benchmark Results from the literature.

**Table 1: Termination on Fixed *MaxGen* (I. E. Fixed Number of Schedules)**

| ESN | Data-Set | Max Schedules | Pad |
|---|---|---|---|
| A2a1 | J30 | 1000 | 5.565 |
| A2a2 | J30 | 5000 | 3.913 |
| A2a3 | J30 | 50000 | 1.919 |
| A2b1 | J60 | 1000 | 22.967 |
| A2b2 | J60 | 5000 | 19.626 |
| A2b3 | J60 | 50000 | 17.338 |

Table 1 displays results when we terminate the Genetic Algorithms after testing fixed number of schedules, i.e. number of generations is kept fixed as is used in the classical form of Genetic Algorithms.

Next we check ability of our 'crude' GA to terminate using Fitness-Deviation Termination. Here we test Fitness-Deviation Termination (FD) with two combinations of ($\mu$, $\delta$ and $\omega$) as depicted in Table 2.

**Table 2: Deviation Limits for Fitness-Deviation Termination**

| Deviations | Limits | |
|---|---|---|
| | Lower | Higher |
| $\sigma_{elites}\ \mu$ | 0.01 | 0.05 |
| $\sigma_{population}\delta$ | 0.10 | 0.50 |
| $\sigma_{generation}\omega$ | 0.10 | 0.50 |

As an additional check, alternately we terminate by keeping *MaxGen* at 100, so we process at most 5000 schedules in any case. The results are displayed as Table 3, where ESN B4a1 and B4b1 belongs to deviation termination criteria set at lower limits, and the other two experiments had the limits set at higher levels.

As evident, a greater number of instances were found to terminate on the *MaxGen* criteria, especially when the deviation limits are kept tight. Moreover, accuracy of the algorithm goes up when the deviation limits are kept tight, as evident from lowering of PAD, as well as higher APLV.

This prompts us to view MaxGen based termination rather than deviation based ones, when we are attempting to optimize problems with a large search area. A related factor against deviation-based termination is possibility of premature termination on reaching the deviation limits, but at a suboptimal location.

**Table 3: Termination by Fitness-Deviation Criteria**

| ESN | Data-Set | Instances Terminating on | | PAD | APLV (This, ESN A) |
|---|---|---|---|---|---|
| | | Deviation Limits | *Max Gen* | | |
| B4a1 | J30 | 826 | 3974 | 3.562 | 8.9701 |
| B4a2 | J30 | 3725 | 1075 | 3.816 | 2.4789 |
| B4b1 | J60 | 18 | 4782 | 17.453 | 11.0720 |
| B4b2 | J60 | 4113 | 687 | 19.761 | -0.6879 |

The Adaptive Termination tests are now carried out, starting with *AdapTerm1*. For our experiments we use a population size (*MaxPop*) of fifty. Therefore, for the J30 data set which has four constrained resources, we shall be testing [30 X 4 X 50 = 6000] schedules. Similarly, for J60 data-set, this figure would be 12000.

**Table 4: Termination by Adaptive Termination 1**

| ESN | Data-Set | Average Schedules (Search Space) | Schedules Tested (Sample Space) | Comparisons | | |
|---|---|---|---|---|---|---|
| | | | | Compared with | PAD | APLV (%) |
| B4a5 | J30 | $5*10^8$ | 6000 | A2a2 | 2.417 | 38.2315 |
| B4b5 | J60 | $7*10^{16}$ | 12000 | A2b2 | 16.169 | 17.6144 |

The results have shown significant improvement, as evident by the more than 38% improvement over our 'primitive' settings. The relatively higher PAD and lower APLV in case of J60 set is due to the increased level of difficulty in optimization of the larger project.

The experimental results for *AdapTerm2* are tabulated in Table 5. We test for two values of logarithmic base, 10 and 20. A higher logarithmic base is seen to act as a better damper. A lower logarithmic base results in higher *MaxGen*, and therefore a large sample area is tested. On scanning through our test results at instance level, it was found that when compared to the total search area (*MaxSdl*), this sample area gets proportionally smaller.

**Table 5: Termination by Adaptive Termination 2**

| ESN | Data Set | Base of Logarithm | PAD | APLV (This, ESN A) |
|---|---|---|---|---|
| B4a8 | J30 | 10 | 1.912 | 51.1372 |
| B4b8 | J60 | | 14.522 | 26.0082 |
| B4a10 | J30 | 20 | 1.994 | 49.0417 |
| B4b10 | J60 | | 14.823 | 24.4733 |

This inverse relationship demonstrates efficiency of our algorithm. In tandem with better efficiency, the PAD inches up favourable on the benchmark result set. The efficiency of the adaptive algorithm is demonstrated in Figure 2, where we plot the Efficiency Index of the instances of the best result, i.e. ESN B4a8. Within the total search space, even though we test a proportionally smaller sample set, but within that we not only improve our PAD rating but also the efficiency level. In other words, as complexity of the Project increases we are able to approach optimal solution set with a proportionally increasing level of efficiency employing a relatively slower increase in *MaxGen* size. For testing Adaptive Termination 3, we employ 10 as the base of logarithm for J30, and for J60 we use 20. For testing ($\phi$ - $\omega$), we use (5-1) and (6-4) for J30 data-set. Since J60 instances have a higher complexity in the range of $10^8$ to $10^{18}$ we test higher values of ($\phi$ - $\omega$), at (10-4) and (10-5). The base of the exponential function is tested at 2 and 1.5.

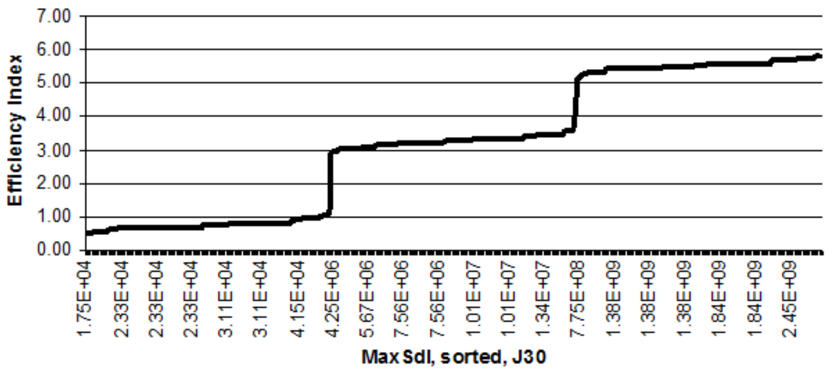

**Figure 2: Efficiency Index vs Project Complexity, Adaptive Termination 2 (Data-Set J30, Sorted on *Maxsdl)***

**Table 6: Experimental Results of Adaptive Termination 3**

| ESN | Data-Set | Adapterm 3 Parameters | Pad | | APLV (This, A 5000) | APLV (This, A 50000) |
|---|---|---|---|---|---|---|
| | | | Benchmark Range | Test Results | | |
| B4a12 | J30 | (10, 2, 5-1) | 0.00 – 2.08 | 0.007 | 99.8195 | 99.6319 |
| B4a14 | | (10, 1.5, 6-0) | | 0.066 | 98.3213 | 96.5770 |
| B4b12 | J60 | (20, 2, 10-4) | 10.71 – 15.94 | 10.812 | 44.9079 | 37.6377 |
| B4b14 | | (20, 1.5, 10-5) | | 11.517 | 41.3179 | 33.5739 |

From the results in Table 6, it is observed that by employing combination of parameter values as described above, the algorithm has achieved the best APLV of 99.8195 for the J30 data-set over results achieved by the initial ('crude') settings. The present result achieved on PAD evaluation is comparable to the best of benchmark results. For the J60 data-set, the best APLV is 44.9079, i.e. almost halfway through towards achieving optimal results. When PAD is compared to benchmark results, the optimal experimental results falls short of the best in literature by only about a percent.

The efficiency index for the best result is provided in Figure 3.

The efficiency of the Adaptive Termination based GA goes up with increase in complexity (or size) of the search space, as depicted in Figure 3. Extremely low PAD and a multifold increase in efficiency is a definite indication of the efficiency of Genetic Algorithms which employ a Damped-Acceleration based Termination criterion.

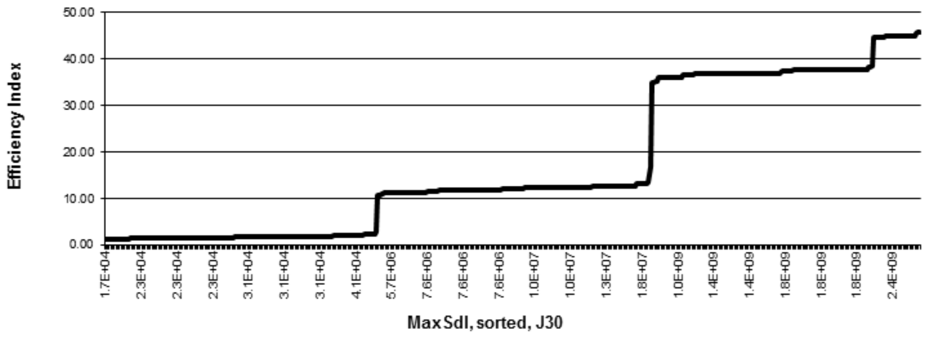

**Figure 3: Efficiency Index vs Project Complexity, Adaptive Termination 3 (Data-Set J30, Sorted on *Maxsdl)***

In Table 7 (a) and 7 (b), the best result are displayed in a tabular format as published in the literature, mostly from Kolisch and Hartmann (2006). The tables are truncated at ten entries, with the topmost being the best. The initial test results of the present work are included at the bottom row of both tables.

**Table 7 (a): Comparison with Benchmark Results: J30 #: Termination Criteria: Max Schedules // Arranged as Sorted on Last Column**

| Sl. No | Author | Year | Algorithm | SGS | Average Deviations % From Optimal Makespan | | |
|---|---|---|---|---|---|---|---|
| | | | | | Termination Criteria # | | |
| | | | | | 1,000 | 5,000 | 50,000 |
| 1 | Ranjbar, et al | 2007 | Scatter Search | Serial | 0.10 | 0.03 | 0.00 |
| 2 | Kochetov, Stolyar | 2003 | GA, TS | Both | 0.10 | 0.04 | 0.00 |
| 3 | Debels, et al | 2006 | Scatter search | Serial | 0.10 | 0.04 | 0.00 |
| 4 | Present Work | 2008 | GA | Serial | 0.007(Adaptive Termination) | | |
| 5 | Kemmoe, et al | 2007 | PSO | | 0.26 | 0.21 | -- |
| 6 | Debels, et al | 2004 | Scatter search | Serial | 0.27 | 0.11 | 0.01 |
| 7 | Valls, et al | 2008 | GA, hybrid | Serial | 0.27 | 0.06 | 0.02 |
| 8 | Valls, et al | 2004 | GA | Serial | 0.34 | 0.20 | 0.02 |
| 9 | Alcaraz et al | 2004 | GA, FB | Both | 0.25 | 0.06 | 0.03 |
| 10 | Alcaraz, Moroto | 2001 | GA, FB | Serial | 0.33 | 0.12 | - - |
| Initial Test Results of Present Work without AdapTerm, etc. | | | | | 5.565 | 3.913 | 1.919 |

For reporting the best results of the present work, we have not segregated it according to the tabled termination criteria. The reason being that our termination criterion is Adaptive Termination 3, which is made flexible proportional to complexity level of the project instance – the very foundation of termination criteria of GA presented in this work.

Moreover, the entry of our best results is kept at a modest distance from the top, by displaying precision of results upto third place after decimal. As evident, we have compared our results only with high-end (50,000) termination criteria entries of other authors. The reason being that Adaptive Termination 3 analyses a proportionally higher number of solutions for complex projects.

**Table 7 (b): Comparison with Benchmark Results: J60 #: Termination Criteria: Max Schedules // Arranged as Sorted on Last Column**

| Sl. No | Author | Year | Algorithm | SGS | Average Deviations % from Optimal Makespan | | |
|---|---|---|---|---|---|---|---|
| | | | | | Termination Criteria # | | |
| | | | | | 1,000 | 5,000 | 50,000 |
| 1 | Ranjbar, et al | 2007 | Scatter Search, FBI | Serial | 11.59 | 11.07 | 10.64 |
| 2 | Debels, et al | 2006 | Scatter search, FBI | Serial | 11.73 | 11.10 | 10.71 |
| 3 | Valls, et al | 2008 | GA, hybrid, FBI | Serial | 11.56 | 11.10 | 10.73 |
| 4 | Kochetov, Stolyar | 2003 | GA, TS | Both | 11.71 | 11.17 | 10.74 |
| 5 | Valls, et al | 2004 | GA, FBI | Serial | 12.21 | 11.27 | 10.74 |

**Table 7 (b): Contd.,**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 6 | Present Work | 2008 | GA | Serial | 10.812 (Adaptive Termination) | | |
| 7 | Alcaraz et al | 2004 | GA, FB, FBI | Both | 11.89 | 11.19 | 10.84 |
| 8 | Hartmann | 2002 | GA, self adapting | Both | 12.21 | 11.70 | 11.21 |
| 9 | Hartmann | 1998 | GA, activity list | Serial | 12.68 | 11.89 | 11.23 |
| 10 | Tormos, Lova | 2003 | Sampling, LFT, FBI | Both | 11.88 | 11.62 | 11.36 |
| Initial Test Results of Present Work without AdapTerm, etc. | | | | | 22.967 | 19.626 | 17.338 |

## CONCLUSIONS

The effectiveness of having an Adaptive Termination for Genetic Algorithms is demonstrated in this work by attempting to optimize the RCPSP, which is a NP-hard problem. Experimental results of the present work show that as compared to initial runs without the Adaptive Termination, the results achieved by gradually tuning the termination criterion are good, with improvement upto 99.6319 %. The best test result of this work is superior to results in the literature where only GA has been employed. The results on using the J60 test instances also has a similar trend, but falls marginally short as compared to results in the literature.

It is expected that results would be far better if a hybrid approach is undertaken – using another metaheuristic for local search once GA locates the optimized solution space. Such hybridization has indeed provided excellent results elsewhere as evident from results of other researchers. The efficiency of using an Adaptive Termination for Genetic Algorithms, when the size of search space may vary, is evident since we achieve higher efficiency level as the search space increases. This increase in efficiency is multifold as we incorporate a Damped-Acceleration based Adaptive Termination.

To a great extent, it is conclusive from this work that by using a Damped-Acceleration based Adaptive Termination criterion in optimizing problems with variable search space, we can achieve a higher level of optimization efficiency of the Genetic Algorithm. The RCPSP was used to prove the concept – which may be adopted for usage in other field as well.

## REFERENCES

1. [Applegate and Cook (1991)] Applegate, D., and W. Cook. A computational study of the job-shop scheduling instance, *ORSA Journal on Computing* 3, 149-156. 1991
2. [CONDOR Report(1988)] Committee on the Next Decade in Operations Research (CONDOR). Operations Research: The Next Decade. *Operations Research* 36:619-637, 1988
3. [Crawford (1996)] Crawford, J M. An Approach to Resource Constrained Project Scheduling. *Proceedings of the 1996 Artificial Intelligence and Manufacturing Research Planning Workshop. Albuquerque, NM., The AAAI Press.* 1996
4. [Kolisch and Hartmann (2006)] Kolisch R. and Hartmann S. Experimental Investigation Of Heuristics For Resource-Constrained Project Scheduling: An Update. *European Journal of Operational Research* 174(1): 23-37. 2006
5. [Muth and Thompson (1963)] Muth, JF, Thompson, GL. *Industrial Scheduling.* Prentice Hall, 1963.
6. [PSPLIB] The Project Scheduling Problems Library, Christian-Albrechts- Universitaet zu Kiel, Germany.
7. [Sarma (2004)] Sarma, TR, Towards an Optimized Algorithm forScheduling and Resource Allocation of Infrastructure Projects; *Managing Trade, Technology and Environment ISBN:81-7446-363-x, Ed. Mallikarjun M and Chugan PK. Excel Books*, 2004

8. [Yang et. al(2001)] Yang, B., Geunes, J., O'Brien, W. J, Resource-Constrained Project Scheduling: Past Work and New Directions, *Research Report 2001-6, Department of Industrial and Systems Engineering, University of Florida.*April 2001

## AUTHOR DETAILS

**Tridib R. Sarma** is an Associate Professor with the School of Management Sciences, Tezpur University, a Central University of India. He received his Bachelor of Engineering in Mechanical Engineering, and MBA from Gauhati University. And earned his PhD in Management from Tezpur University. He has served Oil and Natural Gas Corporation (ONGC) for six years in the capacity as Executive Engineer before moving to Academics in the year 1999. He has first-hand experience of implementing ISO-9002:1994, and is trained in Six-Sigma Green Belt.